Tracing Effects of Covid-19 Over Small and Medium Enterprises

Bonazzi Riccardo
University of Applied Sciences Western Switzerland (HES-SO)
Francesco Maria Cimmino
University of Applied Sciences Western Switzerland (HES-SO)
Jean-Luc Beuchat
University of Applied Sciences Western Switzerland (HES-SO)
Fanny Vérolet
University of Applied Sciences Western Switzerland (HES-SO)

Abstract

In this paper, we describe an ongoing project to assess the liquidity risk of small and medium enterprises (SMEs) in a network. In doing so we try to mimic the tracing applications that have been done to Covid-19. We built a simple artefact under the shape of a method called LUC (Liquidity for Unstructured Collaborations) to (1) collect data that has been encrypted by using multiple keys, (2) store the data in a shared ledger and (3) extract the required information concerning the credit risk of each user while respecting the conditions for the zero-knowledge proof. We see this solution as well-adapted for firms performing unstructured collaboration and we see our service as complementary concerning a trusted company, in the same way, the Covid-19 tracing application collects weak signals and then send patients to doctors for official testing.

Keywords: credit risk, accounting software, blockchain, design science
JEL classification: L10

Paper type: Research article
Received: Jun 2, 2020
Accepted: Aug 12, 2020

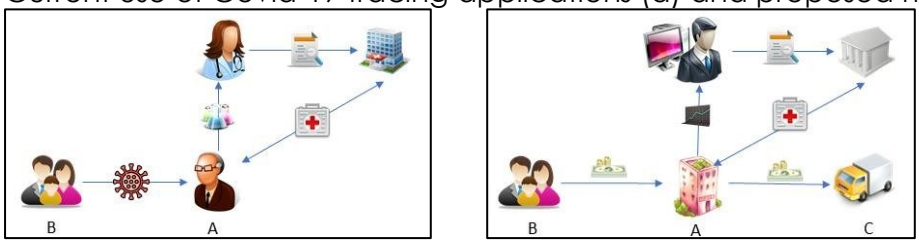
Introduction

This paper is mainly addressed to the design of accounting software and people in charge of supporting business ecosystems. Accordingly, we illustrate the preliminary analysis for an application, which assesses the liquidity risk among firms within a network.

A tracing application to assess the effects of Covid-19 among firms. The pandemic of 2020 had negative health and financial impacts across the world. On the one hand, countries have started developing phone applications to trace the spread of the virus, that informs person (Figure 1) if he comes in contact with a group of people infected "B". Alfred needs to do a test with a doctor, that confirms whether Alfred must be in quarantine and be supported by a health institution. On the other hand, the same application for small and medium enterprises does not exist yet, but figure 1.1.b shows how it could work: if the owner of a firm Alice (A in the figure) comes in contact with clients B that do not pay or suppliers C that do not deliver, the information is sent to a certified accountant in a trusted company, who assesses whether the firms need financial support.

Figure 1

Current use of Covid-19 tracing applications (a) and proposed model for firms (b)



a) b)

Source: Authors' work

Liquidity risk across a network. Most scorecards to assess credit scores mostly focus on data from within the firm (Thomas et al., 2017). Hence, a firm that great financial performance in the past but that his clients, who suddenly cannot pay, can receive a good score, even if it might incur in liquidity problems in the close future. In this paper, we look for a solution to help small and medium enterprise decide whether a potential customer/supplier might have liquidity problems. Indeed, the lack of trust among firms increase their transaction costs (Coase, 1937), it reduces their profits and it might start a negative spiral.

A centralized or decentralized tracing application? Recent events have shown the limitations of centralized applications for emergent solutions. To control the development of COVID-19, many governments have developed contact-tracing apps while others have decided to let users store their data on their phones. For example, Germany and Switzerland had initially chosen the centralized approach but eventually abandoned this approach to use Apple-Google's decentralized system. The reason behind this choice is that important decisions about them seem to be made beyond the reach of democratic governance is worrying for some citizens and it can lower the adoption rate of this type of applications (The Economist, 2020). Therefore, our research question is how to assess the liquidity risk associated with each firm within a network while assuring their privacy?

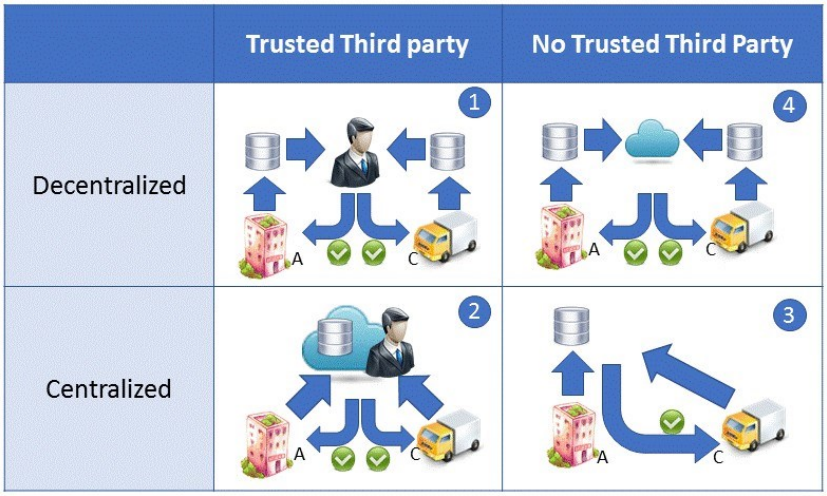
The remainder of the article proceeds as it follows. Section 2 briefly reviews the relevant articles used in this paper. Sections 3 and 4 describe how we created our

artefact and assessed it, whereas section 5 illustrates a simple example. Section 6 concludes the article by discussing its limitations and by proposing new directions of investigation.

Literature review

In this section, we briefly review four different solutions to create a tracing application for SMEs, as shown in Figure 2.

Figure 2
Centralized/Decentralized data management and trusted third-party



Source: Authors' work

Solution 1. SMEs accounting software stored in each firm and shared with the trust company. Most countries impose a procedure in case of over-indebtedness to try to redress the company's situation and thus avoid bankruptcy. For example, Art. 725 of the Swiss Code of Obligations, establishes that measures must be taken without delay if half of the share capital and the legal reserves are no longer covered in the annual balance sheet. Solution 1 represents the baseline case, where each firm has its dataset that is shared once a year with a trusted company, which certifies the data. This solution assesses the risk of each firm, but they update their assessment once a year and to assess credit scores of other companies, a firm needs to buy this data from external companies or compare its performance with industry benchmarks.

Solution 2. SMEs accounting data is shifting to the cloud. Small and medium enterprises are moving towards digitalization, and one could refer to the classification of Lee et al. (2019) to show that there is not a clear option that seems to be adapted for (i) a not-large enterprise, (ii) seeking for an accounting application in the cloud (selective outsourcing), while (iii) paying a fee for service. Nonetheless, there is one solution (referred as "A3/B3"), which is said to deliver economic and strategic benefits for the small firm: the firm will eventually outsource its information technology to multiple providers and it will pay a fee per one use. Thus, we will focus on accounting software based on cloud computing technology. Accordingly, each SME stops downloading accounting software on its computes and pays a monthly fee to access an accounting software online. As a result, accounting data of each SME might be safer in the hands of cloud providers (who can offer state-of-the-art security protocols) and the trust company can offer new

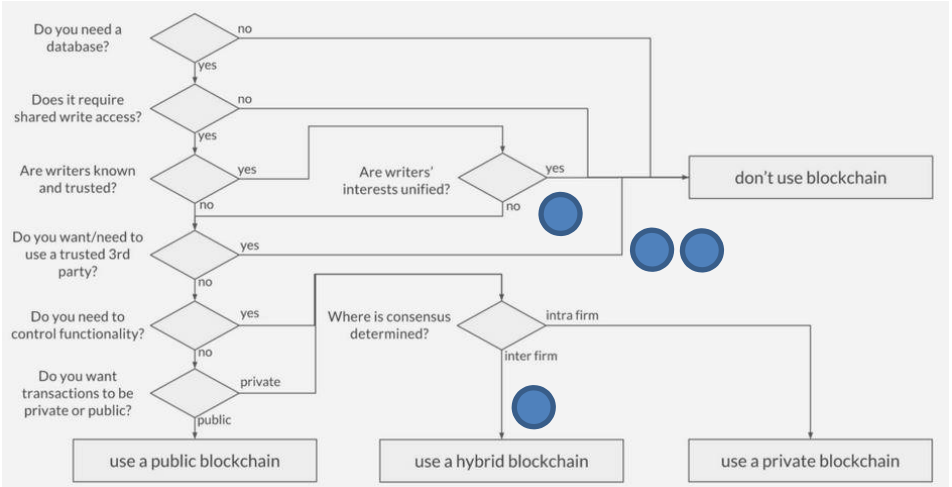
services that present updated results once new data arrives. These services can be defined as microservices: a cohesive, independent process interacting via messages (Dragoni et al., 2017). One example of microservice is a credit scoring of each firm, with the possibility to share such certified score with other firms. It would be like solution 1, but in this case, the score can rapidly change over time.

Solution 3: Sharing data on a centralized database. This type of solution is not used frequently by SMEs, but a big company might impose on its suppliers to use a centralized database. Once we remove the trusted third party, the risk of sharing accounting data with other companies might scare most of them. Nonetheless, a system might allow firms to share the minimum amount of data needed to trust each other. Goldwasserer et al. (1989) propose three properties to describe a so-called zero-knowledge protocol. A simple introduction to zero-knowledge proof protocols is offered by Matthew Green (2014), which offers an interactive zero-knowledge demonstration. Stated informally, they are (1) Completeness: if the system is telling the truth, then they will eventually convince the user (at least with high probability), (2) Soundness: the system can only convince the user if it's telling the truth, and (3) Zero-knowledgeness, meaning that the user does not learn anything else about the software overall solution. This solution would be desirable, in the interests of all users are unified, meaning that all users should agree that the company owning the database should see all the transactions.

Solution 4: Sharing visibility on common data with a decentralized database. Solution 4 is the decentralized solution that most resemble a Covid-19 tracing application. Each user has its dataset and shares only a subset of data with the others, with no trusted third-party. Some accounting software companies have been exploring opportunities linked to distributed ledger technologies (also known as blockchain), and we name the accounting software Banana as an example. In the case of blockchain technology, microservices are offered by smart contracts, as described by Tempesta (2018).

Figure 3

How our solution fits in the decision flow to decide whether to use blockchain



Source: Adapted from (Suichies, 2015)

In recent years, blockchain technology and cryptocurrencies have been a controversial topic in Switzerland. A report published in January (CryptoValley VC, 2020) counts 842 firms in the blockchain ecosystem located between Switzerland and Liechtenstein and the so-called Crypto Valley counts five projects valued at more

than $1B: Ethereum ($14.4B), Dfinity ($2B), Polkadot ($1.2B), Bitmain ($1B), Libra ($1B). That could be a sign of traction, and there are cities in Switzerland like Zug, which allowed citizens to pay part of their taxes with bitcoin, allowing testing the system. Nonetheless, there is a significant gap between early adopters and the adoption of this new technology by the overall population. Indeed, a recent report (Chang, 2020) notices that there have been only 38 Zug citizens that have been paying their taxes with bitcoin since 2016. Hence, it would be worth assessing what is the "job-to-be-done" that each blockchain solution addresses. If we recall the case of Zug, it turns out that some enterprises there have already been accused of illegal operations (Emmel & Pilet, 2019) and that might be a sign that the whole ecosystem needs to reconsider the way it operates. Figure 3 shows how each solution fits in the decision flow to decide whether to use a distributed ledger (Suichies, 2015). Indeed, Greenspan (2015) argues that most of the requirements today are more than fulfilled with relational databases; but not all of them.

The gap in the existing literature. Different solutions seem to be adapted for different contexts, and it might be relevant to think how to reconcile the inner logic of technology for distributed ledger and a basic assumption in the field of organizational economics: if the transaction among two agents is not difficult, then the transaction costs are low and the market is the most efficient solution; instead, if the transaction is difficult the firm has more levers to handle it (Coase, 1937).

In table 1, we list our four solutions and we add three additional columns.

We consider three forms of coordination associated to *effectiveness* (Gibbons, 2003), which depends on the transaction costs associated with the governance structure in a transaction: (L) low for the *partnership* since it requires a significant effort to be maintained, (M) medium for *intermediation* for a trusted third party, and (H) high for a *smart contract* among peers.

Transaction difficulty, that is the collection of features that cause the effectiveness of market governance to decline (Gibbons, 2003): (H) high for partnership among firms, since there are multiple elements to be taken into account, (M) medium for SMEs agreeing to conduct business transactions that involve a significant amount of money or important gaps between cost expenses and received incomes, (L) for SMEs conducting limited exchanges of money that are paid almost immediately.

The last column concerns the context, where each solution outperforms the others:

1. Centralized solutions with no trusted third party might be best suited to governance structures called relational contracts (Gibbons, 2005): is a self-enforcing agreement so rooted in the parties' particular circumstances that the agreement cannot be enforced by a third party.
2. A centralized and decentralized solution involving trusted third-party might be best suited to hybrid governance structures, for example, "contracting for control" (Gibbons, 2005), which means moving control of a part of decision rights from the asset's owner to another party.

- The potential of combining digital identities, smart contracts and transaction in a distributed database might be best suited to something close to market governance called "*unstructured collaboration*" (Baker et al., 2008), which involves separate ownership of assets "a" and "b" by parties "A" and "B"; given the one-shot interaction between the parties, the project will be implemented only when it is in each of the parties' interests to proceed.

Table 1
Each solution is best suited for a specific context and governance structure

Decentralized	3rd party	Effectiveness	Transaction difficulty	Context
No	No	Low (Partnership)	High (Partnership)	Relational contract
No	Yes	Medium (intermediation)	Medium (complex transactions)	Hybrid
Yes	Non	High (smart contract)	Low (small business transactions)	Market (?)
Yes	Yes	Medium (intermediation)	Medium (complex transactions)	Hybrid

Source: Authors' work

Design science methodology

In this section, we illustrate the chosen methodology to answer our research question. We position our study in the field of design science research (Hevner et al., 2004) and we developed an artefact under the shape of a method, as defined by March and Smith (1995). Accordingly, we have followed the guidelines of Gregor and Hevner (2013) to create a set of design principles as part of a nascent design theory, which is at level 2 of the contribution types. We describe the development of our artefact by following the steps of Peffers et al. (2007).

(1) Identify the problem and motivate:

As described in section 1, the focus of our analysis is to create an application for Small and medium enterprises that mimic the Covid-19 tracing app for people. We believe that such a solution will be increasing the trust among firms and reduce the transaction costs and it will help to rapidly track firms that need financial support before the financial damage expands across the network.

(2) Define the objectives of the solution:

As shown in section 2, four types of solutions can be created. In this paper, we will design a decentralized solution without trusted third-party to support unstructured collaboration among small and medium enterprises.

(3) Design and development:

Section 3 illustrates how we conceived a system that traces transactions and assesses the risk of the overall network.

(4) Demonstration:

Section 4 illustrates an example of to design a new service that exploits new trends in the business ecosystem.

(5) Evaluation:

In section 5, we compare the features of the system and across the four solutions.

(6) Communication:

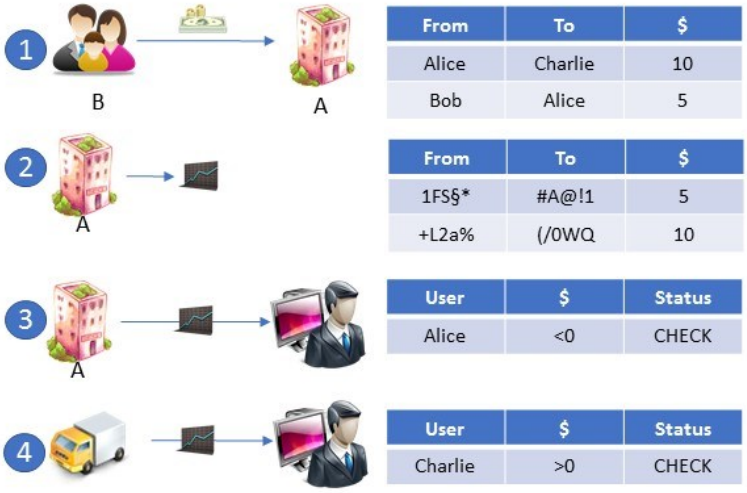
We have started sharing the preliminary results via academic conferences and we plan to submit our full report once the first phase of data collection will be completed.

Our artefact

Figure 4 illustrates the functions of LUC (Liquidity for Unstructured Collaborations). In our example, Alice receives $5 from Bob (client) and sends $10 to Charlie (supplier).

Step 2 shows the first feature of the system: the data is encrypted in a way that does not allow any reader to know who send the money to whom. Then, when Alice sends the data to the microservice, she shares a subset of her transactions and a secret to decrypt the data. In Step 3, the microservice uses the key of Alice to analyse the data and realizes that she has liquidity unbalance of 10 – 5 = USD 5. In step 4, the system shares the information with a certified accountant, who does a detailed analysis. In step 4, Charlie sends the data to the micro-service, who knows that Alice might not pay the money to Charlie. Hence, the system requests a detailed analysis of the certified accountant, even if Charlie does not have a liquidity problem now.

Figure 4.1 Features of the system



Source: Authors' work

Accordingly, we claim that our system shows the following properties:

- Completeness: the micro-service can request to Alice as many subsets of transactions as it needs to be satisfied.
- Soundness: Alice cannot hide transactions to the micro-service and she cannot add fake transactions without having the encryption key of another firm.
- *Zero-knowledgeness*: users cannot trace back the process and get to the original data. Charlie cannot know that Alice has a financial problem; the trust company cannot obtain detailed information from the micro-service; the micro-service does not need to see all the transactions of Alice.

Table 2 compares the three features of the services against the 4 solutions explored in Section 02. In a situation of casual collaboration, a Metamask interface connected with an Ethereum blockchain using Microsoft Azure might assure easier integration and protection against the risk of a super-user having access to the overall dataset.

Table 2
Comparison of features offered by the different solutions

	Microservice	Zero-Knowledge	Effort for client interface
Dec. 3rd party	No	Yes	High
Cen. 3rd party	Yes (often)	Yes	Low (API)
Cen. P2P	Yes (possible)	No	Medium
Dec. P2P	Yes (by design)	Yes	Low (browser)

Source: Authors' work

Illustration example with a simple simulated case

In this section, we present a simple example in 4 steps, to describe how the features of Figure 3 work.

Step A: Assessment of a single firm

In table 3 we show a simplified version of transaction data, which can be obtained by accounting software. Here we take the point of view of the first firm (Alice); hence, if the money is sent to a supplier (Charlie) the amount of the transaction is negative, because Alice is giving money to Charlie instead of receiving it.

Table 3
Simulated transaction data from a hypothetical accounting software

Trans ID	From_ID	To_ID	Amount
1	Bob	Alice	1863
2	Bob	Alice	3241
3	Bob	Alice	9081
4	Charlie	Alice	-6219
5	Darla	Alice	7390

Source: Authors' work

From Table 4 we can extract the information about the expected income and expected costs of each transaction. If a customer does not pay, we list the amount as "not paid". The profit is calculated as Profit = Income – Cost – Not paid. The credit score is calculated as the Credit Score = Total amount paid by customer/ total.

Table 4
Simulated data from a hypothetical accounting software

Trans ID	From_ID	Expected Income	Expected Cost	Not paid	Liquidity
1	Bob	1863	0	0.00	1863.00
2	Bob	3241	0	3241.00	0.00
3	Bob	9081	0	9081.00	0.00
4	Charlie	0	6219	0.00	-6219.00
5	Darla	7390	0	7390.00	0.00

Source: Authors' work

From Table 5 we can assess the liquidity of the firm and assess a score as well, by looking at (Sum of revenues – Sum of Not paid)/(Sum of Costs) = 1863/6219 = 30%.

Table 5
Risk assessment from the dashboard of the financial software

User ID	Revenues	Costs	Not paid	Liquidity
Charlie	0	6219	0	-6219
Bob	14185	0	12322	1863

Source: Authors' work

Step 02: Shared data among multiple firms

In a distributed ledger, each firm can write its transactions. Since we assume that the ledger is public, the ID of each user is encrypted by using a secret. Table 6 shows how the table changes accordingly, even if the financial data in the transaction does not. As we will see, it is possible to pass from Table 6 to Table 9, once we know the criteria, but it is hard to reconstruct Table 6 from Table 9.

The first step is to assign to each user a secret, as shown in Table 8. When two users execute a transaction, each user encrypts its ID using the secret.

Table 6
Each user receives a secret

User_ID	User Secret
Bob	2
Alice	3
Charlie	4

Source: Authors' work

Table 7 shows how the ID is calculated using the formula 100+Transaction ID*Secret. The first three rows are written by Alice and Bob when they execute a transaction. In the first row, the Sender of the first transaction in Table 3 (=Bob) becomes 100 + Transaction ID (=1)+User Secret (=2) = 102. The receiver of the first transaction in Table 6 (=Alice) becomes 100 + Transaction ID (=1) + User secret (=3) = 103.

One could underline that the ID of the sender in the first three rows changes every time, even if the sender is always the same because the data is encrypted.

It is also interesting to notice how the ID of the receiver of row 2 is the same as the id of the sender of row 3. This shows that ID can be the same for different lines, but it occurs randomly.

Table 7
Simulated data on a distributed ledger

Trans ID	From_ID	Expected Income	Expected Cost	Not paid	Liquidity
1	Bob	1863	0	0.00	1863.00
2	Bob	3241	0	3241.00	0.00
3	Bob	9081	0	9081.00	0.00
4	Charlie	0	6219	0.00	-6219.00
5	Darla	7390	0	7390.00	0.00

Source: Authors' work

The added-value of a distributed ledger is row 4, which links Alice with Charlie. This time the amount is positive, and it is written by Alice and Charlie. This means that Alice and Charlie are sharing their lists of transactions on a common database, but it would be hard for Charlie to know that the previous transactions have been executed by Alice.

Step 03: Assessment of liquidity risk for multiple firms

If Bob wants to show to Alice its credit score, he can send his secret to the software. Table 8 shows how the system tries to decrypt all transactions and estimates that Bob has two transactions that are not paid, which are considered as potential infections. The system does not know how much Bob has not paid yet and whether Bob has a liquidity problem, leaving this decision to a certified accountant.

Table 8

Decryption for Bob (secret =2) Simulated results for Bob

T_ID	From_ID	To_ID	Profit
1	102 = 102	103 != 102	1863.00
2	104 = 104	106 != 104	0.00
3	106 = 106	109 ! = 106	0.00
4	101 != 108	112 != 108	-6219.00

T_ID	From_ID	To_ID	Profit
1	Bob		1863.00
2	Bob		0.00
3	Bob		0.00
4			

Source: Authors' work

If Alice wants to show to Charlie her credit score, she can send to the software her secret. The software tries to decrypt the transactions with the secret of Alice. Table 9 shows how the system tries to decrypt all transactions and authenticate those belonging to Alice. The software estimates that Alice has three transactions that are not received, and those are considered as potential infections.

Table 9

Decryption for Alice (secret =3) Simulated results for Alice

T_ID	From_ID	To_ID	Profit
1	102 != 103	103 == 103	1863.00
2	104 != 104	106 == 106	0.00
3	106 != 109	109 == 109	0.00
4	101 != 112	112 == 112	-6219.00

T_ID	From_ID	To_ID	Profit
1		Alice	1863.00
2		Alice	0.00
3		Alice	0.00
4	Alice		6219.00

Source: Authors' work

Step 04: Assessment of each transaction that concerns an individual firm

Table 10 describes a microservice that assesses the risk of the two firms involved in a transaction and that assigns a risk score to each transaction. Rows 2 and 3 show that the liquidity score can change over time, according to new information collected by the micro-service. Such information can be either shared with the users or used only to assess the risk of the network.

Table 10
Micro-service used to create the last column with risks for each transaction

Trans ID	From_ID	Exp. Income	Exp. Cost	Not paid	Liquidity	Score
1	Bob	1863	0	0.00	1863.00	(40%)
2	Bob	3241	0	3241.00	0.00	(20%)
3	Bob	9081	0	9081.00	0.00	(15%)
4	Charlie	0	6219	0.00	-6219.00	(30%)

Source: Authors' work

Discussions and Conclusions

In this paper, we focused on the issue of providing a micro-service to small and medium enterprises for liquidity risk assessment without sharing additional information.

This system aims at creating the required conditions for an efficient market: it provides symmetry of information among users to increase trust and lower transaction costs, whereas it ensures the privacy of companies.

We see this solution as well-adapted for firms performing unstructured collaboration and we see our service as complementary concerning a trusted company, in the same way, the Covid-19 tracing application collects weak signals and then send patients to doctors for official testing.

Accordingly, we built a simple artefact under the shape of a method that (1) collects data that has been encrypted by using multiple keys, (2) stores the data in a shared ledger and (3) extracts the required information concerning the credit risk of each user while respecting the conditions for the zero-knowledge proof.

In its current stage of development, our theoretical model needs to be empirically tested with a functioning prototype and it has just started to describe the complexity of the fields of accounting, information security and organizational economics. Therefore, we intend to collect empirical data of the first version to improve the theoretical model. In the meantime, we identify the following directions of improvement for future research:

1. **Which solution has the lowest cost in terms of operational and capital expenses?** Table 11 compares the different expenses associated with each solution. The decentralized solution with a 3rd party has high costs initially because the infrastructure needs to be replicated in each firm and it has high operational expenses to cover for data integration efforts every year and commission fees for the third party. The creation of a micro-service on a cloud is simple, if the underlying infrastructure is well-conceived, and the only operational expense is the commission fee of the 3rd party. The solution offered by one firm to the other SMEs might cost initially but the infrastructure does not have to be replicated and each transaction costs almost nothing. Finally, the smart contract can be done with a few lines of code, but every transaction will cost money to be written in the distributed dataset (Microsoft Azure, 2020).

Table 11
Comparison of estimated expenses for the different solutions

	Capital Expenditures	Operational Expenditures
Decentralized 3rd party	High	High
Centralized 3rd party	Low	Medium
Centralized Peer-to-Peer (P2P)	Medium	Low
Decentralized Peer-to-Peer (P2P)	Low	Medium

Source: Authors' work

2. **Under which conditions can blockchain technology increase the effectiveness of the system, concerning relational databases?** We have mentioned that blockchain technology allows removing the trusted third party, but if there were not users it would be possible to create alliances among firms to add false information (Gramoli, 2020).
3. **What is the effect of the system over the interaction among firms**? Issues concerning the adaption of the COVID-19 (The Economist, 2020) tracing application will concern also our application. It is important to estimate the critical mass needed to have reliable results and to understand what stops users from using it.

References

1. Baker, G. P., Gibbons, R., Murphy, K. J. (2008), "Strategic alliances: bridges between "islands of conscious power"",, Journal of the Japanese and International Economies, Vol. 22, No. 2, pp. 146-163.
2. Chang, O. (2020), "Bitcoin is a hit with Swiss tax collectors", available at: https://www.cnnmoney.ch/on-the-block/bitcoin-is-a-hit-with-swiss-tax-collectors/ (30 July 2020).
3. Coase, R. (1937), "The theory of the firm", Economica, Vol. 4, No. 16, pp. 386-405.
4. CryptoValley VC. (2020), "CV VC Top 50 Report H2/2019: The blockchain industry in Crypto Valley, Switzerland and Lichtenstein, analyzed and visualized", available at: https://cvvc.com/application/files/8515/7978/3493/CV_VC_Top_50_Report_H2-2019_.pdf (30 July 2020).
5. Dragoni, N., Giallorenzo, S., Lafuente, A. L., Mazzara, M., Montesi, F., Mustafin, R., Safina, L. (2017), "Microservices: yesterday, today, and tomorrow", in Mazzara, M., Meyer, B. (Eds.), Present and Ulterior Software Engineering, Springer, Cham, pp. 195-216.
6. Emmel, C., Pilet, F. (2019), "The dark side of Zug's Crypto Valley", available at: https://www.swissinfo.ch/eng/business/bitfinex-scandal-_the-dark-side-of-zug-s-crypto-valley/45090064 (30 July 2020).
7. Gibbons, R. (2003), "Team theory, garbage cans and real organizations: some history and prospects of economic research on decision-making in organizations", Industrial and Corporate Change, Vol. 12, No. 4, pp. 753-787.
8. Gibbons, R. (2005), "Four formal(izable) theories of the firm?", Journal of Economic Behavior & Organization, Vol. 58, No. 2, pp. 200-245.
9. Goldwasser, S., Micali, S., Rackoff, C. (1989), "The knowledge complexity of interactive proof systems", SIAM Journal on Computing, Vol. 18, No. 1, pp. 186-208.
10. Gramoli, V. (2020), "From blockchain consensus back to Byzantine consensus", Future Generation Computer Systems, Vol. 107, pp. 760-769.
11. Green, M. (2014), "Zero knowledge proofs: an illustrated primer", available at: https://blog.cryptographyengineering.com/2014/11/27/zero-knowledge-proofs-illustrated-primer/ (14 August 2020).

12. Greenspan, G. (2015), "Avoiding the pointless blockchain project: How to determine if you've found a real blockchain use case', available at: https://www.multichain.com/blog/2015/11/avoiding-pointless-blockchain-project/ (30 July 2020).
13. Gregor, S., Hevner, A. R. (2013), "Positioning and presenting design science research for maximum impact", MIS Quarterly, Vol. 37, No. 2, pp. 337-355.
14. Hevner, A. R., March, S. T., Park, J., Ram, S. (2004), "Design science in information systems research", MIS Quarterly, Vol. 28, No. 1, pp. 75-105.
15. Lee, Y. C. (2019), "Adoption intention of cloud computing at the firm level", Journal of Computer Information Systems, Vol. 59, No. 1, pp. 61-72.
16. March, S. T., Smith, G. F. (1995), "Design and natural science research on information technology", Decision Support Systems, Vol. 15, No.), pp. 251-266.
17. Microsoft Azure. (2020), "Pricing - blockchain service", available at: https://azure.microsoft.com/en-us/pricing/details/blockchain-service/ (16 August 2020).
18. Peffers, K., Tuunanen, T., Rothenberger, M. A., Chatterjee, S. (2007), "A design science research methodology for information systems research", Journal of Management Information Systems, Vol. 24, No. 3, pp. 45-77.
19. Suichies, B. (2015), "Why blockchain must die in 2016", available at: https://medium.com/block-chain/why-blockchain-must-die-in-2016-e992774c03b4 (15 August 2020).
20. Tempesta, S. (2018), "Microservices - architect blockchain applications as microservices", available at: https://docs.microsoft.com/en-us/archive/msdn-magazine/2018/september/microservices-architect-blockchain-applications-as-microservices (15 August 2020).
21. The Economist. (2020), "App-based contact tracing may help countries get out of lockdown", available at: https://www.economist.com/science-and-technology/2020/04/16/app-based-contact-tracing-may-help-countries-get-out-of-lockdown (15 August 2020).
22. Thomas, L., Crook, J., Edelman, D. (2017), Credit Scoring and Its Applications (2nd ed.), SIAM Publications, Philadelphia.

About the authors

Riccardo Bonazzi is a professor at the Institute of entrepreneurship and management of the HES-SO Valais Wallis. His main interests are business model design and organization design. The author can be contacted at **riccardo.bonazzi@hevs.ch.**

Francesco Maria Cimmino is a PhD student at the institute of entrepreneurship and management of the HES-SO Valais. His main interests are energy management and machine learning algorithms. The author can be contacted at **francesco.cimmino@hevs.ch.**

Jean-Luc Beuchat is a professor at the institute of information systems of the HES-SO Valais Wallis. His main interests are computer arithmetic, cryptography and computer architecture. The author can be contacted at **jeanluc.beuchat@hevs.ch.**

Fanny Vérolet is a research assistant at the Institute of entrepreneurship and management of the HES-SO Valais. Her main interests are accounting and business model design. The author can be contacted at **fanny.verolet@hevs.ch.**